

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
)
ARNOLD et al.) Group Art Unit: (Not Yet Assigned)
)
Serial No.: (Not Yet Assigned)) Examiner: (Not Yet Assigned)
)
Filed: March 16, 2001)
)
For: METHOD AND APPARATUS FOR)
DYNAMIC DISTRIBUTED)
COMPUTING OVER A NETWORK)
)

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

PRELIMINARY AMENDMENT

Prior to the examination of the above application, please amend this application as follows:

IN THE ABSTRACT:

Please delete the Abstract of the Disclosure and replace it with the following:

--A homogeneous execution environment operates within a heterogeneous client-server network. A client selects a server and transmits a procedure call with parameters. In response, a server dynamically and securely downloads code to a compute server; invokes a generic compute method; executes the code on the compute server; and returns the results to the calling client method, preserving the result on the compute server if requested. This technique is efficient in that it does not require multiple copies of code to be downloaded or compiled since server byte-

codes can be executed on each of the different systems, therefore downloading or compiling multiple copies of code can be avoided. The code can be compiled once and downloaded as needed to the various servers as byte-codes and then executed.--

IN THE CLAIMS:

Please cancel claims 1 and 2, substitute clean amended claims 3, 9, 11-12, 17 and 18 for the pending claims with the same number, and add claims 28-52 as follows:

3. (Amended) A method performed on a processor operatively coupled to a collection of servers which enables a client associated with the processor to dynamically distribute a task to a server, the method comprising the steps of:

selecting a server to process the task;

forming a task request from parameters and data;

sending the task request to the selected server which downloads any needed executable byte code, invokes a generic compute technique capable of executing the task request on the selected server and generates results; and

receiving the results back from the selected server.

9. (Amended) The method of claim 3, wherein selecting the server comprises selecting the server based on the overall processing load distribution among the collection of servers.

11. (Amended) The method of claim 3, wherein selecting the server comprises selecting the server based on the specialized computing capabilities of each server.

12. (Amended) The method of claim 11, wherein the specialized computing capabilities include a capability to render images.

17. (Amended) The method of claim 3, wherein the results comprise an object.

18. (Amended) A method performed on a processor operatively coupled to a collection of servers which enables a server associated with the processor to dynamically receive and process a task from a client computer wherein the task is in an executable programming language compatible with each of the server computers, the method comprising the steps of:

retrieving parameters and data from a task request into a task;

downloading any needed executable byte code;

invoking a generic compute method on the server, which is capable of processing a plurality of types of tasks, which executes the task and generates results; and

returning results to the client.

--28. A computer readable medium containing instructions for controlling a computer system comprising a collection of servers to perform a method for enabling a client to dynamically distribute a task to a server, the method comprising the steps of:

selecting a server to process the task;

forming a task request from parameters and data;

sending the task request to the selected server which downloads any needed executable byte code, invokes a generic compute method capable of executing the task request on the

selected server and generates results; and

receiving the results back from the selected server.

29. The computer readable medium of claim 28, wherein the computer system is operatively coupled to a primary storage device, a secondary storage device, a display device, and an input/output mechanism.

30. The computer readable medium of claim 28, wherein the task is developed in a programming language and environment compatible with each of the servers.

31. The computer readable medium of claim 28, wherein the selected server is selected from a plurality of heterogeneous computer systems.

32. The computer readable medium of claim 30, wherein the environment includes a remote procedure call subsystem.

33. The computer readable medium of claim 32, wherein the remote procedure call subsystem is the Remote Method Invocation (RMI) system.

34. The computer readable medium of claim 28, wherein selecting the server comprises selecting the server based on the overall processing load distribution among the collection of servers.

35. The computer readable medium of claim 28, wherein selecting the server comprises selecting the server based on a lowest load characteristic compared to an average load characteristic of the servers over a predetermined period of time.

36. The computer readable medium of claim 28, wherein selecting the server comprises selecting the server based on the specialized computing capabilities of each server.

37. The computer readable medium of claim 36, wherein the specialized computing capabilities include a capability to render images.

38. The computer readable medium of claim 28, wherein the sending step further comprises:
determining whether code related to the requested task is present on the selected server;
and
downloading the code onto the selected server if the code is not present on the selected server.

39. The computer readable medium of claim 28, wherein the sending step further comprises:
providing the task as a parameter to the generic compute method.

40. The computer readable medium of claim 28 further comprising the step of indicating to the server that results from a computed task should be stored in a result cache on the selected server for subsequent tasks to use.

41. The computer readable medium of claim 28, wherein the results are used for further processing on the client.

42. The computer readable medium of claim 28, wherein the results comprise an object.

43. A computer readable medium containing instructions for controlling a computer system comprising a collection of servers to perform a method for enabling a server to dynamically receive and process a task from a client computer wherein the task is in an executable programming language compatible with each of the servers, the method comprising the steps of:

retrieving parameters and data from a task request into a task;

downloading any needed executable byte code;

invoking a generic compute method on the server, which is capable of processing a plurality of types of tasks, which executes the task and generates results; and

returning results to the client.

44. The computer readable medium of claim 43, wherein the computer system is operatively coupled to a primary storage device, a secondary storage device, a display device, and an input/output mechanism.

45. The computer readable medium of claim 43, wherein the task is developed in a programming language compatible with each of the servers.

46. The computer readable medium of claim 43, wherein the task is developed using a Java programming language and environment.

47. The computer readable medium of claim 43, wherein the environment includes a remote procedure call subsystem.

48. The computer readable medium of claim 47, wherein the remote procedure call subsystem is the Remote Method Invocation (RMI) system.

49. The computer readable medium of claim 43, wherein the retrieving step further comprises:

determining if types related to the task are available on the server;

when the types are not available on the server, downloading the types onto the server from a location as indicated by the parameters provided by the client; and

executing the task based upon the data and parameters provided by the client.

50. The computer readable medium of claim 49, wherein the determining step and the downloading steps are performed by a remote procedure call (RPC) subsystem.

51. The computer readable medium of claim 50, wherein the determining step is performed by a Remote Method Invocation (RMI) type of remote procedure call subsystem.

52. The computer readable medium of claim 43, further comprising:
storing the results from the task in a cache if a subsequent task will use the results.

REMARKS

Applicants respectfully request that the Abstract of the Disclosure and the Claims of the instant application be amended as provided herein. Applicants submit that the claims as amended recite a combination not taught or suggested by the prior art.

In view of the foregoing amendments and remarks, Applicants respectfully request the consideration and examination of this application and the timely allowance of the pending claims.

If there is any fee due in connection with the filing of this Preliminary Amendment, please charge the fee to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: March 15, 2001

By: 

Jason E. Gorden
Reg. No. 46,734

MARKED-UP CLAIMS

3. (Amended) A method performed on a processor operatively coupled to a collection of servers which enables a client associated with the processor to dynamically distribute a task to a server, the method comprising the steps of:

selecting a server to process the task;

forming a task request from [the] parameters and data;

sending the task request to the selected server which downloads any needed executable byte code, invokes a generic compute technique capable of executing the task request on the selected server and generates results; and

receiving the results back from the selected server.

9. (Amended) The method of claim 3, wherein [a criteria for selection the server includes] selecting the server comprises selecting the server based on the overall processing load distribution among the collection of [server computers] servers.

11. (Amended) The method of claim 3, wherein [a criteria for selection the server includes] selecting the server comprises selecting the server based on the specialized computing capabilities of each server [computer].

12. (Amended) The method of claim 11, wherein the specialized computing capabilities [includes rendering] include a capability to render images.

17. (Amended) The method of claim 3, wherein the [result is] results comprise an object.

18. (Amended) A method performed on a processor operatively coupled to a collection of servers which enables a server associated with the processor to dynamically receive and process a task from a client computer wherein the task is in an executable programming language compatible with each of the server computers, the method comprising the steps of:

retrieving parameters and data from a task request into a task;

downloading any needed executable byte code;

invoking a generic compute method on the server, which is capable of processing a plurality of types of tasks, which executes the task and generates results; and

returning results to the client.